



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/822,454

04/12/2004

Erez Haba

13768.1433

2824

47973 7590 12/08/2009
WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER

WANG, BEN C

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

12/08/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/822,454	Applicant(s) HABA ET AL.	
	Examiner BEN C. WANG	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 14 September 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-5, 11, 12, 17, 18, 20, 22, 23, 25 and 27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-5, 11, 12, 17, 18, 20, 22, 23, 25, and 27 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Applicant's amendment dated September 14, 2009, responding to the Office Action mailed August 28, 2009 provided in the rejection of claims 1-5, 11, 12, 17, 18, 20, 22, 23, 25, and 27, where claims 1 (spelling out the acronyms "XML" and "XSLT") and 12 (correcting the typographic error) have been amended.

Claims 1-5, 11, 12, 17, 18, 20, 22, 23, 25, and 27 remain pending in the application and which have been fully considered by the examiner.

The status of claims 1 and 12 were inadvertently listed as "Previously Presented" instead of —Currently Amended—

Applicant's arguments with respect to claims currently amended have been fully considered but are not persuasive. Please see the section of "Response to Arguments" for details.

Applicant's arguments with respect to claims rejection under 35 USC § 103(a) are not persuasive, thus the previous rejections are maintained and reproduced below.

2. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Response to Arguments

3. Applicant's arguments filed on September 14, 2009 have been fully considered but they are not persuasive.

In the remarks, Applicant argues that, for examples:

(A.1) The examiner has filed to show that the following limitations are taught by the art: "a test case file component that receives metadata that defines which versions of the one or more test cases test which versions of the source under test, and stores the metadata in an Extensible Markup Language (XML) file in conjunction with test results that are generated by executing the one or more test cases on the source under test, wherein metadata is also stored which indicates the version of the one or more test cases and the version of the source under test to which the test results correspond, the test case file component further storing attributes in the XML file that enable the querying of the test results;" (stated in 2nd full-paragraph on page 10 in Remarks – emphasis added)

Art Unit: 2192

(A.2) Neither does reference provide any indication of how it could be modified to provide both benefits simultaneously (stated in 2nd full paragraph on page 9 through 1st non-full paragraph on page 10)

(A.3) *Mandava* does not maintain versioning (stated in 2nd full paragraph on page 9 – emphasis added)

Examiner's response:

(R.1) As per the argument one (A.1) above, firstly McNeely clearly teaches a test case file component that receives metadata that defines which versions of the one or more tests test which versions of the source under test, and stores the metadata in conjunction with test results that are generated by executing the one or more tests on the source under test, wherein metadata is also stored which indicates the version of the one or more tests and the version of the source under test to which the test results correspond (e.g., [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case (*interpreted as versions of test case*), which may in turn specify a particular program or software version (*interpreted as version of the source under test*) that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment (*interpreted as binding the version of test with the*

Art Unit: 2192

version of the source under test into corresponding versioned test results) – emphasis added);

Secondly, *Mandava-2* clearly discloses the test case file component further storing attributes in the XML file that enables the querying of the test results (e.g., see the exhibit A below, elements 118” – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 18, Lines 55 - 67 - ... a merge component 138 can be implemented to merge the uniform test results in the first dynamic XML file 114”a ... into a merged dynamic XML result file 118” stored in storage 116; Col. 19, Lines 10 - 24 - ... Thus, the reporter tool (*interpreted as querying of the test results*) of the present invention is very powerful and flexible as the reporter tool eliminates the necessity to manually ...; Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case identification, a status of the test case, and a status description (*interpreted as storing attributes in the XML file*) – emphasis added)

Thirdly, McNeely discloses binding the version of test with the version of the source under test into corresponding versioned test results and further *Mandava-2* teaches using stored XML to enable the query of the test result as explained above. Indeed, the combined system from both *McNeely* and *Mandava-2* clearly teaches combined system of the first type and second type and provides both of these benefits using a single XML file as explained above (stated in 2nd paragraph of page 8 through 1st paragraph of page 9 in Remarks - emphasis added).

Thus, the combined teachings from both *McNeely* and *Mandava-2* clearly meet the portion of claim limitations as stated in A.1.

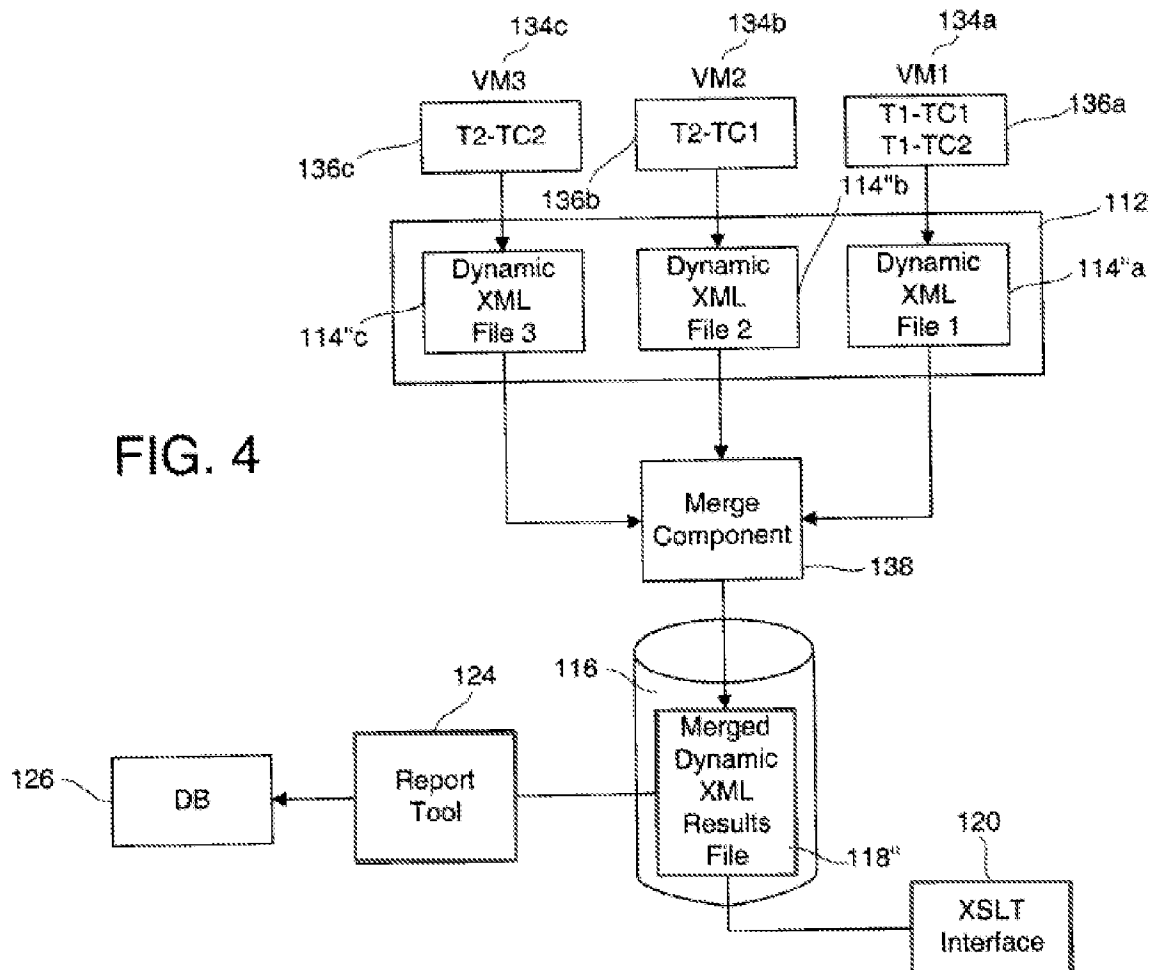


Exhibit A – excerpted from Figure 4 in the prior art of Mandava-2

(R.2) As per the argument two (A.2) above, firstly *McNeely* teaches the present invention includes a fully integrated test case and test plan editor where test plans and their associated test cases are maintained in a version-controlled environment (e.g. see

paragraph [0017] – emphasis added); the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test); as such, a test plan may specify a particular version of a test case, which may in turn specify a particular program of software version that is to be installed on a given DUT for the duration of test (see paragraph [0019] - emphasis added)

Secondly *Mandava-2* teaches maintaining versioning (e.g., see Col. 10, TABLE 1 – entry 5 – Add Test for the TestSuite using one version of add Test – In one example, i.e. addTest(String testSuiteID, String testId, String testName, String testDescription), addTest(String testSuiteID, String testId, String test Name) ...; entry 7 - Set Status for TestCase using one version of setTestCaseStatus - ... testCaseID ... - emphasis added) and executing a test suite and generating uniform test results; also including storing the uniform test results so as to allow viewing of the uniform test result (e.g., see Abstract; also see the Exhibit A above for using single XML file - emphasis added)

Thus, both arts are analogous art and related to version aware test management system and method.

(R.3) As per the argument three (A.3) above, *Mandava-2* clearly teaches maintaining versioning (e.g., see Col. 10, TABLE 1 – entry 5 – Add Test for the TestSuite using one version of add Test – In one example, i.e. addTest(String testSuiteID, String testId, String testName, String testDescription), addTest(String testSuiteID, String testId, String test Name) ...; entry 7 - Set Status for TestCase using one version of setTestCaseStatus - ... testCaseID ... - emphasis added)

Claim Objections

4. Claims 1, 17, and 23 are objected to because the following informalities:
- "... that enable the querying ...", in claim 1 at line 13, should be corrected to read -- ... that enables the querying ... --, as to overcome the typographic error.
 - Acronyms "XML" and "XSLT" should be spelled out at the first appearance in claims 17 and 23.

Appropriate correction is required (See MPEP § 608.01(b))

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1, 2, 5, 12, 17, 18, 22, 23, 25, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over McNeely et al. (Pub. No. US 2002/0162059 A1) (hereinafter 'McNeely') in view of Mandava (Pat. No. US 7,203,928 B2) (hereinafter 'Mandava-2')
6. **As to claim 1** (Currently Amended), McNeely discloses an application test management system that maintains fine-grained versioning of tests and their

Art Unit: 2192

relationship to software under test without sacrificing querying, filtering, and reporting, the system comprising:

a computer readable storage medium having stored thereon the following components executable by a processor:

a version component that detects versions of a source under test and versions of one or more tests that test the source under test:

- a test case file component that receives metadata that defines which versions of the one or more tests test which versions of the source under test, and stores the metadata in conjunction with test results that are generated by executing the one or more tests on the source under test, wherein metadata is also stored which indicates the version of the one or more tests and the version of the source under test to which the test results correspond (e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment; [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled

environment; [0039]; [0060] - ... test plan and test case manager 316

examines information contained in a test case or test plan related to the
operating software version required on a particular test device ...;

NOTE: [0019] – The version-controlled environment may be extended to
include operating software associated with each DUT (Device Under Test).

As such, a test plan may specify a particular version of a test case
(interpreted as versions of test case), which may in turn specify a particular
program or software version *(interpreted as version of the source under test)*
that is to be installed on a given DUT for the duration of the test.

Corresponding test results may also be stored and maintained in the version-
controlled environment *(interpreted as binding the version of test with the*
version of the source under test into corresponding versioned test results) –
emphasis added)

Further, McNeely discloses the present invention includes a fully integrated test case and test plan editor where test plans and their associated test cases are maintained in a version-controlled environment (e.g. see paragraph [0017] – emphasis added); the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test); as such, a test plan may specify a particular version of a test case, which may in turn specify a particular program of software version that is to be installed on a given DUT for the duration of test (see paragraph [0019] - emphasis added) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Results*, Mandava-2 discloses:

- storing the metadata in an Extensible Markup Language (XML) file (e.g., Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case identification, a status of the test case, and a status description);
- the test case file component further storing attributes in the XML file that enable(s) the querying of the test results (e.g., Col. 8, Lines 46-50 - ... the static XML file is configured to include entries for each and every test and test case ... the static XML file also includes a comment describing the function of each test case and test;

NOTE: see the exhibit A below, elements 118” – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 18, Lines 55 - 67 - ... a merge component 138 can be implemented to merge the uniform test results in the first dynamic XML file 114”a ... into a merged dynamic XML result file 118” stored in storage 116; Col. 19, Lines 10 - 24 - ... Thus, the reporter tool (*interpreted as querying of the test results*) of the present invention is very powerful and flexible as the reporter tool eliminates the necessity to manually ...; Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case identification, a status of the test case, and a status description (*interpreted as storing attributes in the XML file*) – emphasis added); and

- a component that uses the attributes of the XML file to transform the XML file utilizing Extensible Stylesheet Language Transformations (XSLT) to enable the

querying of the test results based on the version of the source under test and the version of the one or more tests which correspond to the test results (e.g., Fig. 4, elements 118” – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 8, Lines 17-25 - ... the uniform results are stored to storage 116 in a dynamic XML result file 118. The uniform results in the dynamic XML 118 can be viewed by a user 122 using a, Extensible Stylesheet Language (XSLT) Stylesheet interface 120; **NOTE:** McNeely teaches the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment (e.g., [0019]))

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the McNeely’s system to further provide other limitations stated above in the McNeely system.

The motivation is that it would further enhance the McNeely’s system by taking, advancing and/or incorporating the Mandava-2’s system which offers significant advantages for executing a test suite and generating uniform test results; also including storing the uniform test results so as to allow viewing of the uniform test result as once suggested by Mandava-2 (e.g., see Abstract; also see the Exhibit A below for using single XML file - emphasis added)

7. **As to claim 2** (Previously Presented) (incorporating the rejection in claim 1), McNeely discloses the system wherein the attributes includes a pointer to the source under test (e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment; [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test ...)

8. **As to claim 5** (Previously Presented) (incorporating the rejection in claim 1), McNeely discloses the system wherein the attributes include a pointer to a test (e.g., [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment)

9. **As to claim 12** (Currently Amended) (incorporating the rejection in claim 11), McNeely discloses the system wherein the test results are generated by a test execution component that executes the one or more tests on the source under test

Art Unit: 2192

(e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment; [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment; [0039]; [0060] - ... test plan and test case manager 316 examines information contained in a test case or test plan related to the operating software version required on a particular test device ...)

10. **As to claim 17** (Previously Presented), Foster discloses a test management methodology comprising:

- retrieving metadata that defines a version of source code and a version of one or more test cases that test the source code (e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment);
- persisting the metadata in conjunction with test results that are generated by executing the one or more tests on the source code, wherein metadata is also

stored which indicates the version of the one or more tests and the version of the source code to which the test results correspond (e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment; [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment; [0039]; [0060] - ... test plan and test case manager 316 examines information contained in a test case or test plan related to the operating software version required on a particular test device ...;

NOTE: [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case (*interpreted as versions of test case*), which may in turn specify a particular program or software version (*interpreted as version of the source under test*) that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment (*interpreted as binding the*

version of test with the version of the source under test into corresponding versioned test results) – emphasis added);

Further, McNeely discloses the present invention includes a fully integrated test case and test plan editor where test plans and their associated test cases are maintained in a version-controlled environment (e.g. see paragraph [0017] – emphasis added); the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test); as such, a test plan may specify a particular version of a test case, which may in turn specify a particular program of software version that is to be installed on a given DUT for the duration of test (see paragraph [0019] - emphasis added) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Results*, Mandava-2 discloses:

- persisting the metadata to an XML file (e.g., Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case identification, a status of the test case, and a status description);
- the test case file component further storing attributes in the XML file that enable the querying of the test results (e.g., Col. 8, Lines 46-50 - ... the static XML file is configured to include entries for each and every test and test case ... the static XML file also includes a comment describing the function of each test case and test;

NOTE: see the exhibit A below, elements 118” – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 18, Lines 55 - 67 - ... a merge component 138 can be implemented to merge the uniform test results in the first dynamic XML file 114”a ... into a merged dynamic XML result file 118” stored in storage 116; Col. 19, Lines 10 - 24 - ... Thus, the reporter tool (*interpreted as querying of the test results*) of the present invention is very powerful and flexible as the reporter tool eliminates the necessity to manually ...; Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case identification, a status of the test case, and a status description (*interpreted as storing attributes in the XML file*) – emphasis added); and

- transforming the XML file utilizing XSLT and the attributes to enable a user to view at least one of exception patterns, trends, productivity, and success rates and management operations including at least one of selection, query, reporting, suit composition, and scheduling (e.g., Fig. 4, elements 118” – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 8, Lines 17-25 - ... the uniform results are stored to storage 116 in a dynamic XML result file 118. The uniform results in the dynamic XML 118 can be viewed by a user 122 using a, Extensible Stylesheet Language (XSLT) Stylesheet interface 120; **NOTE:** McNeely teaches the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given

DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment (e.g., [0019]))

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the McNeely's system to further provide other limitations stated above in the McNeely system.

The motivation is that it would further enhance the McNeely's system by taking, advancing and/or incorporating the Mandava-2's system which offers significant advantages for executing a test suite and generating uniform test results; also including storing the uniform test results so as to allow viewing of the uniform test result as once suggested by Mandava-2 (e.g., see Abstract; also see the Exhibit A below for using single XML file - emphasis added)

11. **As to claim 18** (Previously Presented) (incorporating the rejection in claim 17), McNeely discloses the method wherein the metadata that defines the versions of the source code and the one or more tests is retrieved from a version component that monitors changes to the source code and the one or more tests (e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment; [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn

Art Unit: 2192

specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment; [0039]; [0060] - ... test plan and test case manager 316 examines information contained in a test case or test plan related to the operating software version required on a particular test device ...)

12. **As to claim 22** (Original) (incorporating the rejection in claim 17), please refer to claim **17** above, accordingly.

13. **As to claim 23** (Previously Presented), Foster discloses a testing methodology comprising:

- loading a test case in accordance with a test case file stored in a source file;
- executing the test case on a source code under test;
- generating test results, wherein the test results are version tagged to indicate the relationships between test results, version of the test case, and version of the source code under test (e.g., Fig. 4, elements 302 – System Under Test ; 316 – Test Plan/Case Execution Manager; 318 – Report Manager; 350 – Version Control Environment; 352 – Test Plan/Case Library; 354 – Test Results Library; [0017] - ... test plans and their associated test cases are maintained in a version-controlled environment; [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed

on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment; [0039]; [0060] - ... test plan and test case manager 316 examines information contained in a test case or test plan related to the operating software version required on a particular test device ...;

NOTE: [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case (*interpreted as versions of test case*), which may in turn specify a particular program or software version (*interpreted as version of the source under test*) that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment (*interpreted as binding the version of test with the version of the source under test into corresponding versioned test results*) – emphasis added);

Further, McNeely discloses the present invention includes a fully integrated test case and test plan editor where test plans and their associated test cases are maintained in a version-controlled environment (e.g. see paragraph [0017] – emphasis added); the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test); as such, a test plan may specify a particular version of a test case, which may in turn specify a particular program of software version that is to be installed on a given DUT for the duration of test (see

Art Unit: 2192

paragraph [0019] - emphasis added) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Results*, Mandava-2 discloses:

- saving the test results to an XML file, wherein the XML file stores metadata that defines the version of the source code and the version of the test case which were executed to generate the test results, and wherein the XML file further stores pointers to the version of the source code and the version of the test case (e.g., Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case identification, a status of the test case, and a status description; Col. 8, Lines 46-50 - ... the static XML file is configured to include entries for each and every test and test case ... the static XML file also includes a comment describing the function of each test case and test'

NOTE: see the exhibit A below, elements 118" – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 18, Lines 55 - 67 - ... a merge component 138 can be implemented to merge the uniform test results in the first dynamic XML file 114"a ... into a merged dynamic XML result file 118" stored in storage 116; Col. 19, Lines 10 - 24 - ... Thus, the reporter tool (*interpreted as querying of the test results*) of the present invention is very powerful and flexible as the reporter tool eliminates the necessity to manually ...; Col. 4, Lines 24-30 - ... the dynamic XML file may include a test case

identification, a status of the test case, and a status description (*interpreted as storing attributes in the XML file*) – emphasis added); and

- employing XSLT to transform the XML file into an in memory representation of a database that enables the test results to be queried (e.g., Fig. 4, elements 118” – Merged Dynamic XML Results File; 120 – XSLT Interface; 124 – Report Tool; Col. 8, Lines 17-25 - ... the uniform results are stored to storage 116 in a dynamic XML result file 118. The uniform results in the dynamic XML 118 can be viewed by a user 122 using a, Extensible Stylesheet Language (XSLT) Stylesheet interface 120; **NOTE:** McNeely teaches the version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test. Corresponding test results may also be stored and maintained in the version-controlled environment (e.g., [0019]))

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the McNeely's system to further provide other limitations stated above in the McNeely system.

The motivation is that it would further enhance the McNeely's system by taking, advancing and/or incorporating the Mandava-2's system which offers significant advantages for executing a test suite and generating uniform test results; also including storing the uniform test results so as to allow viewing of the uniform test result as once

Art Unit: 2192

suggested by Mandava-2 (e.g., see Abstract; also see the Exhibit A below for using single XML file - emphasis added)

14. **As to claim 25** (Original) (incorporating the rejection in claim 23), Mandava-2 discloses the method further comprising publishing the test results to an enterprise data store (e.g., Fig. 3A, elements 116 and 118; Col. 8, Lines 17-26 - ... the uniform results are stored to storage 116 in a dynamic XML result file 118)

15. **As to claim 27** (Original) (incorporating the rejection in claim 23), please refer to claim **23** above, accordingly.

16. Claims 3, 4, 11 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over McNeely in view of Mandava-2 and Mandava (Pat. No. US 7,210,066 B2) (hereinafter 'Mandava-1')

17. **As to claim 3** (Previously Presented) (incorporating the rejection in claim 1), McNeely and Mandava-2 do not explicitly disclose the limitations stated below.

However, in an analogous art of *Method and System for Determining Computer Software Test Coverage*, Mandava-1 discloses:

- the system wherein the attributes include a pointer to requirement for test data (e.g., Fig. 1A; Col. 2, Lines 30-46 - ... Each assertion document has a corresponding tagged assertion for each assertion in the respective specification. Each tagged assertion is defined in a markup language ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-1 into the McNeely-Mandava-2's system to further provide the limitations stated above in the McNeely-Mandava-2 system.

The motivation is that it would further enhance the McNeely-Mandava-2's system by taking, advancing and/or incorporating the Mandava-1's system which offers significant advantages for allowing assertions in a specification document to be correlated with data in a static XML; and a user can query the assertion coverage tool of the present invention so as to determine whether a specific assertion in the specification document has been tested, or whether a specific assertion has been tested in excess by a plurality of test cases as once suggested by Mandava-1 (e.g., Col. 28, Lines 42-61)

18. **As to claim 4** (Previously Presented) (incorporating the rejection in claim 1), Mandava-1 discloses the system wherein the attributes include a pointer to requirement (e.g., Fig. 1A; Col. 2, Lines 30-46 - ... Each assertion document has a corresponding tagged assertion for each assertion in the respective specification. Each tagged assertion is defined in a markup language ...; Col. 2, Lines) and McNeely discloses configuration under test data (e.g., [0090] - ... device configuration information ...)

19. **As to claim 11** (Previously Presented) (incorporating the rejection in claim 1), Mandava-1 discloses the system wherein the XML file is stored in a catalog with other XML files, and wherein the XML file has a hierarchical relationship with at least one of

Art Unit: 2192

the other XML files (e.g., Figs. 3E-1; 3E-2; 3F-1; Col. 26, Line 61 through Col. 27, Line 22 - ... a test suite structure ...)

20. **As to claim 20** (Previously Presented) (incorporating the rejection in claim 17), McNeely discloses the method wherein the attributes comprises a pointer to at least one of the source code (e.g., [0019] – The version-controlled environment may be extended to include operating software associated with each DUT (Device Under Test). As such, a test plan may specify a particular version of a test case, which may in turn specify a particular program or software version that is to be installed on a given DUT for the duration of the test), further Mandava-1 discloses a requirement under test (e.g., Fig. 1A; Col. 2, Lines 30-46 - ... Each assertion document has a corresponding tagged assertion for each assertion in the respective specification. Each tagged assertion is defined in a markup language ...), and furthermore McNeely discloses a configuration under test (e.g., [0090] - ... device configuration information ...)

Conclusion

21. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

Art Unit: 2192

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/

Ben C. Wang

Examiner, Art Unit 2192

/Michael J. Yigdall/

Primary Examiner, Art Unit 2192